

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

T. Okamura et al.  
11/26/03  
Q78646  
10f1

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2 0 0 2 年 1 2 月 3 日

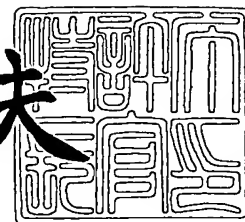
出 願 番 号  
Application Number: 特 願 2 0 0 2 - 3 5 0 8 4 6  
[ST. 10/C]: [ J P 2 0 0 2 - 3 5 0 8 4 6 ]

出 願 人  
Applicant(s): 日 本 電 気 株 式 会 社

2 0 0 3 年 9 月 1 2 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 3 - 3 0 7 5 3 3 1

【書類名】 特許願

【整理番号】 35001183

【あて先】 特許庁長官殿

【国際特許分類】 G06F 11/10

【発明者】

    【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

    【氏名】 岡村 利彦

【発明者】

    【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

    【氏名】 穴田 啓晃

【特許出願人】

    【識別番号】 000004237

    【氏名又は名称】 日本電気株式会社

【代理人】

    【識別番号】 100079005

    【弁理士】

    【氏名又は名称】 宇高 克己

【手数料の表示】

    【予納台帳番号】 009265

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

    【包括委任状番号】 9715827

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 誤り訂正符号復号装置

【特許請求の範囲】

【請求項 1】 低密度パリティ検査符号のメッセージ・パッシング復号方式に基づく復号装置であって、

受信値および前記復号方式の過程で発生するメッセージを格納する複数のメモリ手段と、

前記復号方式における複数の変数ノード関数手段と、

前記復号方式における複数のチェックノード関数手段と、

前記メモリ手段のアドレスを発生する複数のアドレス発生手段と、

前記変数ノード関数手段と前記チェックノード関数手段との接続を定める複数のシャッフルネットワーク手段と  
を有し、

前記アドレス発生手段は複数の置換に基づいて前記アドレスを発生し、前記シャッフルネットワーク手段はそれぞれ一部の前記変数ノード関数手段と接続され、かつ該接続は複数の置換に基づいて決定され、復号過程において前記アドレス発生手段における前記置換の変更と前記シャッフルネットワーク手段における前記置換の変更とは同一周期で行うことを特徴とする誤り訂正符号復号装置。

【請求項 2】 前記アドレス発生手段は 1 個で、すべての前記メモリ手段に対してアドレスを発生し、

前記シャッフルネットワーク手段は 1 個で、すべての前記変数ノード関数手段と接続されることを特徴とする請求項 1 に記載の誤り訂正符号復号装置。

【請求項 3】 前記メモリ手段は前記メッセージとともにその和を格納することを特徴とする請求項 1 又は請求項 2 に記載の誤り訂正符号復号装置。

【請求項 4】 前記アドレス発生手段をカウンタで実現することを特徴とする請求項 1 から請求項 3 のいずれかに記載の誤り訂正符号復号装置。

【請求項 5】 前記シャッフルネットワーク手段の置換をガロア体演算に基づき決定することを特徴とする請求項 1 から請求項 4 のいずれかに記載の誤り訂正符号復号装置。

【請求項 6】 前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数手段の出力であるメッセージを 1 より小さい係数を乗じて補正することを特徴とする請求項 1 から請求項 5 のいずれかに記載の誤り訂正符号復号装置。

【請求項 7】 前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数手段は入力メッセージの絶対値の最小値とそのインデックス、および入力メッセージの 2 番目の最小値および入力メッセージの正負の情報を保持することを特徴とする請求項 1 から請求項 6 のいずれかに記載の誤り訂正符号復号装置。

【請求項 8】 前記アドレス発生手段のみを変更することで異なる符号の復号に対応することを特徴とする請求項 1 から請求項 7 のいずれかに記載の誤り訂正符号復号装置。

【請求項 9】 前記アドレス発生手段および前記シャッフルネットワーク手段の一つに対応する前記変数ノード関数手段の組に対して、その出力において符号語ビットが非常に高い確率で 0 であるというメッセージを常に送信する機能を設けることで不均一低密度パリティ検査符号の復号を実現することを特徴とする請求項 1 から請求項 8 のいずれかに記載の誤り訂正符号復号装置。

【請求項 1 0】 コンピュータに、低密度パリティ検査符号のメッセージ・パッシング復号方式に基づく復号を行わせるプログラムであって、前記プログラムは、コンピュータを、

前記復号方式における複数の変数ノード関数手段と、

前記復号方式における複数のチェックノード関数手段と、

受信値および前記復号方式の過程で発生するメッセージを格納する複数のメモリ手段のアドレスを、複数の置換に基づいて発生するアドレス発生手段と、

前記アドレス発生手段の置換の変更と同一周期で変更される置換に基づいて、変数ノード関数手段とチェックノード関数手段との接続を定めるシャッフルネットワーク手段として機能させることを特徴とするプログラム。

【請求項 1 1】 前記メモリ手段に前記メッセージとともにその和が格納されていることを特徴とする請求項 1 0 に記載のプログラム。

【請求項 12】 前記プログラムは、前記シャッフルネットワーク手段における置換をガロア体演算に基づき決定することを特徴とする請求項 10 又は請求項 11 に記載のプログラム。

【請求項 13】 前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数手段の出力であるメッセージを 1 より小さい係数を乗じて補正することを特徴とする請求項 10 から請求項 12 のいずれかに記載のプログラム。

【請求項 14】 前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数手段は入力メッセージの絶対値の最小値とそのインデックス、および入力メッセージの 2 番目の最小値および入力メッセージの正負の情報を保持することを特徴とする請求項 10 から請求項 13 のいずれかに記載のプログラム。

【請求項 15】 前記アドレス発生手段の機能のみを変更することで異なる符号の復号に対応することを特徴とする請求項 10 から請求項 14 のいずれかに記載のプログラム。

【請求項 16】 前記アドレス発生手段および前記シャッフルネットワーク手段の一つに対応する前記変数ノード関数手段の組に対して、その出力において符号語ビットが非常に高い確率で 0 であるというメッセージを常に送信する機能を設けることで不均一低密度パリティ検査符号の復号を実現することを特徴とする請求項 10 から請求項 15 のいずれかに記載のプログラム。

【請求項 17】 低密度パリティ検査符号のメッセージ・パッシング復号方式に基づく復号方法であって、

受信値および前記復号方式の過程で発生するメッセージが格納されたメモリのアドレスを、複数の置換に基づいて発生するステップと、

前記アドレス発生の変換の変更と同一周期で変更される置換に基づいて、前記復号方式における複数の変数ノード関数と前記復号方式における複数のチェックノード関数とを結びつけるステップと  
を有することを特徴とする誤り訂正符号復号方法。

【請求項 18】 前記メモリには、メッセージとともにその和が格納されていることを特徴とする請求項 17 に記載の訂正符号復号方法。

【請求項 19】 変数ノード関数とチェックノード関数とを結びつけは、ガロア体演算に基づき決定することを特徴とする請求項 17 又は請求項 18 に記載の誤り訂正符号復号方法。

【請求項 20】 前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数の出力であるメッセージを 1 より小さい係数を乗じて補正することを特徴とする請求項 17 から請求項 19 のいずれかに記載の誤り訂正符号復号方法。

【請求項 21】 前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数は入力メッセージの絶対値の最小値とそのインデックス、および入力メッセージの 2 番目の最小値および入力メッセージの正負の情報を保持することを特徴とする請求項 17 から請求項 20 のいずれかに記載の誤り訂正符号復号方法。

【請求項 22】 メモリのアドレスの発生を変更することにより、異なる符号の復号に対応することを特徴とする請求項 17 から請求項 21 のいずれかに記載の誤り訂正符号復号方法。

#### 【発明の詳細な説明】

##### 【0001】

#### 【発明の属する技術分野】

本発明は誤り訂正符号化および復号処理に関し、特に低密度パリティ検査符号 (Low-Density Parity-Check Code, 以下、LDPC符号と略す) に対する復号装置に関する。

##### 【0002】

#### 【従来の技術】

誤り訂正符号は符号化、復号という処理によってデータ伝送時に混入するノイズの影響を低減する技術である。符号化は送信するデータに冗長を付加する処理であり、符号化されたデータを符号語と呼ぶ。通信路に送出された符号語はノイズの影響を受けて受信時には符号語の一部のビットが反転するといったエラーが起こる。復号はこのエラーの影響を受けた受信語から冗長を利用してデータを復元する処理である。

## 【 0 0 0 3 】

LDPC符号は1960年代に提案された誤り訂正符号であるが、ターボ符号との関連が指摘され、1990年代後半になって注目を集めるようになった（例えば、非特許文献1参照。）。

## 【 0 0 0 4 】

## 【非特許文献1】

D.J.C.Macky and R.M.Neal, "Good error correcting codes based on very sparse matrices," IEEE Transactions on Information Theory 45 pp.399-431, 1999

## 【 0 0 0 5 】

誤り訂正符号は、一般に推定処理である復号処理が符号化処理に比べて複雑となる。特に、訂正能力の高い大きな符号長の符号で最尤復号、もしくはそれに近い復号性能を出す場合には非常に複雑度が高い処理が必要となる。

## 【 0 0 0 6 】

LDPC符号は、非常に0の多いパリティ検査行列を持つ、ということで特徴付けられ、この性質から比較的効率の良い高性能復号法が存在している。この復号方法はメッセージ・パッシング復号方式（Message-Passing decoding, 以下、MP復号と略す）と呼ばれている。

## 【 0 0 0 7 】

以下、LDPC符号のMP復号について説明を行う。尚、簡単のために符号はバイナリであり、変調方式はBPSKでエラーなしの場合は符号語のビット0に対しては+1、ビット1には-1という軟判定値が受信値となるとする。

## 【 0 0 0 8 】

LDPC符号におけるMP復号は、パリティ検査行列に対応するタナーグラフと呼ばれる2分グラフを用いて説明される。図14にパリティ検査行列Hと対応するタナーグラフGの例を示す。

## 【 0 0 0 9 】

タナーグラフGのノードは、変数ノードとチェックノードとの2種類に分類される。変数ノードは行列Hの列、つまり符号語ビットに対応する。チェックノード

ドは行列Hの行に対応する。これに対して行列Hで1が立っている関係にあるノード間をエッジで結ぶことによって対応するグラフが構成され、このグラフを行列Hに対するタナグラフと呼ぶ。

#### 【0010】

LDPC符号の復号法はグラフのエッジに割り当てられた“メッセージ”と呼ばれる量をノードにおいて反復的に更新することで実行される。メッセージはチェックノードから変数ノードと、変数ノードからチェックノードとの2種類が存在し、それぞれノードで計算された符号語ビットの信頼度情報に相当する。

#### 【0011】

変数ノード、チェックノードでのアルゴリズムについては幾つかの方法が知られている。

#### 【0012】

最も復号特性が良い方法の一つとしてsum-productアルゴリズムがある。ここでは比較的複雑度の小さいmin-sumアルゴリズムとよばれる方法について説明する。

#### 【0013】

ある変数ノードに対応する符号語ビットの受信値を $r$ 、この変数ノードへのチェックノードからのメッセージを $c(1), (2), \dots, c(d_v)$ とおく ( $d_v$ はこの変数ノードから出ているエッジの個数)。この変数ノードは $c(1)$ に対応するエッジの終点のチェックノードに対しては[式1]で表される $v(1)$ を送出する。

$$[\text{式1}] \quad v(1) \leftarrow r + c(2) + \dots + c(d_v)$$

$v(i)$  ( $i=2, \dots, d$ )についても同様に $r$ と $c(j)$  ( $j \neq i$ ) との和として算出する。

#### 【0014】

一方、あるチェックノードへのメッセージを $v(1), \dots, v(d_c)$ としたとき ( $d_c$ はこのチェックノードから出ているエッジの個数)、このチェックノードは $v(1)$ に対応するエッジの終点の変数ノードに対して[式2]で表される $c(1)$ を送出する。



$$[式2] \quad c(1) \leftarrow \text{sgn}(v(2), \dots, v(d_c)) \cdot \min\{|v(2)|, \dots, |v(d_c)|\}$$

ここで $\text{sgn}(v(2), \dots, v(d_c))$ は $v(i)$  ( $i=2, \dots, d_c$ ) の正負の符号(+1または-1)を乗じた値であり、 $|a|$ は $a$ の絶対値、 $\min$ は最小値を選択する関数である。

【0015】

$c(i)$  ( $i=2, \dots, d$ ) についても同様に $v(j)$  ( $j \neq i$ ) を用いて算出する。

【0016】

LDPC符号ではノードあたりのエッジの個数が小さいため、[式1]、[式2]の処理の複雑度は小さくて済む。

【0017】

すべてのノードに対して一通り[式1]及び[式2]を実行する処理の単位を1 iteration処理と呼ぶことにする。MP復号はこの1 iteration処理を繰り返すことで遂行される。繰り返し回数は通常10から30回程度である。

【0018】

最終的な0, 1の判定(硬判定)は各符号語ビットにおいて、[式3]の正負を判定することによって行われる。

$$[式3] \quad r+c(1)+\dots+c(d_v)$$

この硬判定結果がすべてのチェックノードを満たした時点でMP復号の反復処理は終了する。

【0019】

グラフG全体をそのまま装置化できれば高速化が期待できるが、LDPC符号は通常長い符号長(1000~)で使用するため、これは困難となる。そこで、メッセージをメモリに格納し、回路を共有してグラフGのノードを部分的に並列に動作させて装置複雑度とスループットの調節を図ることになる。

【0020】

この方針に沿って、はじめに部分的に並列に動作する復号器を設計し、それに

合致する符号を構成する符号化方式設計がある（例えば、非特許文献 2）。

【 0 0 2 1 】

【非特許文献 2】

E. Bautillon, J. Castura, and F. R. Kschischang, “Decoder-First Code Design”, the 2nd International Symposium on Turbo Codes and Related Topics, pp. 459-462, 2000

【 0 0 2 2 】

図 1 5 は非特許文献 2 に開示された復号器の構成図である。

【 0 0 2 3 】

図 1 5 の復号器について説明する。

【 0 0 2 4 】

メモリ手段 1 5 0 1 は受信値および[式 2]のメッセージ  $c(i)$  を保持し、図 1 5 の復号器ではこれを複数個持つ。

【 0 0 2 5 】

アドレス発生手段 1 5 0 2 は、メモリ手段 1 5 0 1 にアクセスするアドレスを発生し、メモリ手段 1 5 0 1 と 1 対 1 に対応している。

【 0 0 2 6 】

変数ノード関数手段 1 5 0 3 は、[式 1]の処理を実行する回路であり、メモリ手段 1 5 0 1 と 1 対 1 に対応している。

【 0 0 2 7 】

チェックノード関数手段 1 5 0 4 は、[式 2]の処理を実行する回路である。入出力の個数はタナグラフにおけるチェックノードのエッジの個数と一致している。

【 0 0 2 8 】

シャッフルネットワーク手段 1 5 0 5 は、変数ノード関数手段 1 5 0 3 とチェックノード関数手段 1 5 0 4 との接続を定める。

【 0 0 2 9 】

この復号器では個々の変数ノード関数手段 1 5 0 3 は、メモリ手段 1 5 0 1 と 1 対 1 に対応しているため、メモリ手段 1 5 0 1 へのアクセスの衝突なしに並列

に動作させることが可能である。

#### 【0 0 3 0】

なおかつ、このとき個々のチェックノード関数への全入力が一斉に得られるようになっているため、チェックノード関数手段 1 5 0 5 も並列に動作させることができる。

#### 【0 0 3 1】

このように図 1 5 の復号器を用いて効率よく部分的な並列処理を実現できる。

#### 【0 0 3 2】

##### 【発明が解決しようとする課題】

しかしながら、図 1 5 の復号器はその構成が最適化されているわけではなく、より簡易な構成にできる可能性がある。このとき、符号性能の劣化がないようにしなければならない。

#### 【0 0 3 3】

また、非特許文献 2 ではアドレス発生手段やシャッフルネットワーク手段を具体的にどのように与えるのか述べられていない。

#### 【0 0 3 4】

さらに、図 1 5 の構成ではシャッフルネットワークがすべての変数ノード関数手段の出力を置換する形態となっているため、実装時のリソースの都合でさらに小さな部分で並列化を行うときには処理を制御するスケジュールが複雑になる恐れがあった。

#### 【0 0 3 5】

そこで、本発明は上記課題に鑑みて発明されたものであって、その目的は LDPC 符号の復号装置において従来装置と比べて簡易で、かつ柔軟度の高い並列復号器を提供することにある。

#### 【0 0 3 6】

##### 【課題を解決するための手段】

本発明の目的を達成する第 1 の発明は、低密度パリティ検査符号のメッセージパッシング復号方式に基づく復号装置であって、

受信値および前記復号方式の過程で発生するメッセージを格納する複数のメ

メモリ手段と、

前記復号方式における複数の変数ノード関数手段と、

前記復号方式における複数のチェックノード関数手段と、

前記メモリ手段のアドレスを発生する複数のアドレス発生手段と、

前記変数ノード関数手段と前記チェックノード関数手段との接続を定める複数のシャッフルネットワーク手段と

を有し、

前記アドレス発生手段は複数の置換に基づいて前記アドレスを発生し、前記シャッフルネットワーク手段はそれぞれ一部の前記変数ノード関数手段と接続され、かつ該接続は複数の置換に基づいて決定され、復号過程において前記アドレス発生手段における前記置換の変更と前記シャッフルネットワーク手段における前記置換の変更とは同一周期で行うことを特徴とする。

#### 【0037】

本発明の目的を達成する第2の発明は、上記第1の発明において、前記アドレス発生手段は1個で、すべての前記メモリ手段に対してアドレスを発生し、

前記シャッフルネットワーク手段は1個で、すべての前記変数ノード関数手段と接続されることを特徴とする。

#### 【0038】

本発明の目的を達成する第3の発明は、上記第1又は第2の発明において、前記メモリ手段は前記メッセージとともにその和を格納することを特徴とする。

#### 【0039】

本発明の目的を達成する第4の発明は、上記第1、第2又は第3の発明において、前記アドレス発生手段をカウンタで実現することを特徴とする。

#### 【0040】

本発明の目的を達成する第5の発明は、上記第1、第2、第3又は第4の発明において、前記シャッフルネットワーク手段の置換をガロア体演算に基づき決定することを特徴とする。

#### 【0041】

本発明の目的を達成する第6の発明は、上記第1、第2、第3、第4又は第5

の発明において、前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数手段の出力であるメッセージを1より小さい係数を乗じて補正することを特徴とする。

#### 【0042】

本発明の目的を達成する第7の発明は、上記第1、第2、第3、第4、第5又は第6の発明において、前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数手段は入力メッセージの絶対値の最小値とそのインデックス、および入力メッセージの2番目の最小値および入力メッセージの正負の情報を保持することを特徴とする。

#### 【0043】

本発明の目的を達成する第8の発明は、上記第1、第2、第3、第4、第5、第6又は第7の発明において、前記アドレス発生手段のみを変更することで異なる符号の復号に対応することを特徴とする。

#### 【0044】

本発明の目的を達成する第9の発明は、上記第1、第2、第3、第4、第5、第6、第7又は第8の発明において、前記アドレス発生手段および前記シャッフルネットワーク手段の一つに対応する前記変数ノード関数手段の組に対して、その出力において符号語ビットが非常に高い確率で0であるというメッセージを常に送信する機能を設けることで不均一低密度パリティ検査符号の復号を実現することを特徴とする。

#### 【0045】

本発明の目的を達成する第10の発明は、コンピュータに、低密度パリティ検査符号のメッセージ・パッシング復号方式に基づく復号を行わせるプログラムであって、前記プログラムは、コンピュータを、

前記復号方式における複数の変数ノード関数手段と、

前記復号方式における複数のチェックノード関数手段と、

受信値および前記復号方式の過程で発生するメッセージを格納する複数のメモリ手段のアドレスを、複数の置換に基づいて発生するアドレス発生手段と、

前記アドレス発生手段の置換の変更と同一周期で変更される置換に基づいて、

変数ノード関数手段とチェックノード関数手段との接続を定めるシャッフルネットワーク手段として機能させることを特徴とする。

【0046】

本発明の目的を達成する第11の発明は、上記第10の発明において、前記メモリ手段に前記メッセージとともにその和が格納されていることを特徴とする。

【0047】

本発明の目的を達成する第12の発明は、上記第10又は第11の発明において、前記プログラムが、前記シャッフルネットワーク手段における置換をガロア体演算に基づき決定することを特徴とする。

【0048】

本発明の目的を達成する第13の発明は、上記第10、第11又は第12の発明において、前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数手段の出力であるメッセージを1より小さい係数を乗じて補正することを特徴とする。

【0049】

本発明の目的を達成する第14の発明は、上記第10、第11、第12又は第13の発明において、前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数手段は入力メッセージの絶対値の最小値とそのインデックス、および入力メッセージの2番目の最小値および入力メッセージの正負の情報を保持することを特徴とする。

【0050】

本発明の目的を達成する第15の発明は、上記第10、第11、第12、第13又は第14の発明において、前記アドレス発生手段の機能のみを変更することで異なる符号の復号に対応することを特徴とする。

【0051】

本発明の目的を達成する第16の発明は、上記第10、第11、第12、第13、第14又は第15の発明において、前記アドレス発生手段および前記シャッフルネットワーク手段の一つに対応する前記変数ノード関数手段の組に対して、その出力において符号語ビットが非常に高い確率で0であるというメッセージを

常に送信する機能を設けることで不均一低密度パリティ検査符号の復号を実現することを特徴とする。

#### 【0052】

本発明の目的を達成する第17の発明は、低密度パリティ検査符号のメッセージ・パッシング復号方式に基づく復号方法であって、

受信値および前記復号方式の過程で発生するメッセージが格納されたメモリのアドレスを、複数の置換に基づいて発生するステップと、

前記アドレス発生 of 置換の変更と同一周期で変更される置換に基づいて、前記復号方式における複数の変数ノード関数と前記復号方式における複数のチェックノード関数とを結びつけるステップとを有することを特徴とする。

#### 【0053】

本発明の目的を達成する第18の発明は、上記第17の発明において、前記メモリには、メッセージとともにその和が格納されていることを特徴とする。

#### 【0054】

本発明の目的を達成する第19の発明は、上記第17又は第18の発明において、変数ノード関数とチェックノード関数とを結びつけは、ガロア体演算に基づき決定することを特徴とする。

#### 【0055】

本発明の目的を達成する第20の発明は、上記第17、第18又は第19の発明において、前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数の出力であるメッセージを1より小さい係数を乗じて補正することを特徴とする。

#### 【0056】

本発明の目的を達成する第21の発明は、上記第17、第18、第19又は第20の発明において、前記復号方式はmin-sumアルゴリズムに基づき、前記チェックノード関数は入力メッセージの絶対値の最小値とそのインデックス、および入力メッセージの2番目の最小値および入力メッセージの正負の情報を保持することを特徴とする。

**【0057】**

本発明の目的を達成する第22の発明は、上記第17、第18、第19、第20又は第21の発明において、メモリのアドレスの発生を変更することにより、異なる符号の復号に対応することを特徴とする。

**【発明の実施の形態】**

次に、本発明の実施の形態について図面を参照して詳細に説明する。

**【0058】**

図1を参照すると、本発明は受信値およびMP復号に必要なメッセージを格納するメモリ手段101と、メモリ手段101にアクセスするアドレスを発生するアドレス発生手段102と、MP復号における変数ノードの処理を行う変数ノード関数手段103と、MP復号におけるチェックノードの処理を行うチェックノード関数手段104と、変数ノード手段103とチェックノード手段104とを接続するシャッフルネットワーク手段105とを含む。

**【0059】**

これらの構成要素の概略を図1に基づいて述べる。以下、K及びDは復号器によって決定される整数で、kは1からK、dは1からDまでの任意の整数をとるものとする。

**【0060】**

メモリ手段101は、符号語ビットの組に対応する $M(1,1)$ から $M(K,D)$ までのDK個のメモリから成るメモリ群で、変数ノードに対応する受信値および変数ノードへの入力となるメッセージを格納する。エッジの本数が大きく、メッセージの量が多い場合は $M(k,d)$ も複数のメモリを用いて実現する場合もある。

**【0061】**

アドレス発生手段102は、 $AG(1)$ から $AG(K)$ までのK個の回路から成る。

**【0062】**

$AG(k)$ は同一値域を持つ複数の置換に基づきアドレスを発生し、 $M(k,1)$ から $M(k,D)$ までのD個のメモリに対してアドレスを発生する。各 $AG(k)$ は複数の置換を周期的に使用する。尚、 $AG(k)$ はリアルタイムでアドレスを生成する回路として実現する場合もあるし、メモリ（ROM）を利用して実現する場合もある。



## 【0063】

変数ノード関数手段103はMP復号における変数ノード処理を実行するVN(1,1)からVN(K,D)までのDK個の回路から成る。各VN(k,d)はメモリ手段102のM(k,d)から受信値、メッセージを読み込み、チェックノード関数手段の入力となる一つの出力を生成する。また、VN(k,d)はメモリ手段M(k,d)への書き込みを制御する機能も持つ。この機能は別途手段を用意して対処することも可能である。

## 【0064】

チェックノード関数手段104は、MP復号におけるチェックノード処理を実行するCN(1)からCN(D)までのD個の回路から成る。複数の変数ノード生成手段103で生成されたメッセージを読み込み、それぞれの変数ノードの入力となるメッセージを生成する。生成されたメッセージはメモリ手段101に格納される。

## 【0065】

シャッフルネットワーク手段105は、変数ノード関数手段103とチェックノード関数手段104とを連結するSN(1)からSN(K)までのK個の回路から成る。SN(k)はVN(k,1)からVN(k,D)とCN(1)からCN(D)のそれぞれ1つの入出力インターフェースが1対1に対応するように接続する。各SN(k)は集合{1,2,...,D}上の複数の置換を用いて実現される。

## 【0066】

次に、図2、3、4、5を参照して本実施の形態の全体の動作について詳細に説明する。

## 【0067】

図2は本発明の復号装置の動作の概略を示したフローチャートである。

## 【0068】

まず、受信値を用いてメモリ手段101を初期化する(ステップ201)。

## 【0069】

次に、アドレス発生手段102、シャッフルネットワーク手段105の設定を行う(ステップ202)。

## 【0070】

続いて、設定されたアドレス発生手段102、シャッフルネットワーク手段1

05の下でメモリ手段101、変数ノード関数手段103およびチェックノード関数手段104を動作させてメッセージ更新処理を行う（ステップ203）。

#### 【0071】

次にMP復号における1 iteration処理が終了したかどうかを判断する（図2のステップ204）。

#### 【0072】

終了していない場合には、ステップ202に戻りアドレス発生手段102、シャッフルネットワーク手段105を次の設定としてメッセージ更新処理を行う。

#### 【0073】

ステップ204で1 iteration処理が終了している場合には、復号処理を終了するかどうか判断する（図2のステップ205）。この終了判定は硬判定結果がパリティ検査行列のすべてのチェックを満たすかどうかで判断する。硬判定結果はメモリ手段101に格納する方法も、別のメモリ手段を用意して格納する方法も考えられる。

#### 【0074】

ステップ205で終了と判定された場合には、復号結果を出力して復号処理を終了する（図2のステップ206）。終了と判定されなかった場合には、ステップ201に戻り、アドレス発生手段102、シャッフルネットワーク手段105を1 iteration処理の開始時の設定として復号処理を継続する。

#### 【0075】

図3、4はステップ202から205の処理の流れを詳しく示した図である。

#### 【0076】

図3は本発明の復号装置で適用されるLDPC符号のパリティ検査行列の構成を示している。

#### 【0077】

パリティ検査行列301は置換行列 $R(j, k)$ を $J \times K$ 個並べるパリティ検査符号の構成を示している。パリティ検査行列のブロック $R(j, k)$ の構造式302は $R(j, k)$ が更に二つの置換行列 $Q(j, k)$ と $P(j, k)$ のクロネッカー積で表されていることを示している。 $R(j, k)$ は $Q(j, k)$ の要素が1の場合には $P(j, k)$ , 0の場合には $P(j, k$

)と同じ大きさの零行列で置き換えることで定義される行列である。

【0 0 7 8】

図 1 のDはP(j, k)の大きさに対応している。Q(j, k) が大きさEのとき、R(j, k)はDEの大きさの置換行列となる。

【0 0 7 9】

図 4 は図 3 のパリティ検査行列に対して、アドレス発生手段 1 0 2 およびシャッフルネットワーク手段 1 0 5 の個数Kが図 3 の検査行列の列ブロックの個数と一致している場合のステップ 2 0 2 から 2 0 5 までの処理フローについて、特にステップ 2 0 2 について詳述した流れ図である。

【0 0 8 0】

まず、パリティ検査行列の行ブロックのインデックスjを初期値(j=1)に設定する(ステップ 4 0 1)。

【0 0 8 1】

次に、アドレス発生手段 1 0 2 であるAG(1)からAG(K)をQ(j, 1)からQ(j, K)に、シャッフルネットワーク手段 1 0 5 であるSN(1)からSN(K)をP(j, 1)からP(j, K)に設定する(ステップ 4 0 2)。

【0 0 8 2】

メッセージ更新処理のステップ 4 0 3 は後で詳述する。

【0 0 8 3】

次に、jを1インクリメントしてj=J+1となったら1 iteration処理終了と判定する(ステップ 4 0 4、4 0 5)。

【0 0 8 4】

1 iteration処理が終了していないときにはステップ 4 0 2 に戻る。そうでないときにはステップ 2 0 5 と同一に終了判定をする(ステップ 4 0 6)。

【0 0 8 5】

終了と判定されなかったときにはステップ 4 0 1 に戻り新たなiteration処理を開始する。

【0 0 8 6】

図 5 は図 4 のステップ 4 0 3 のメッセージ更新処理を説明する図である。以下

、 $k$ は1から $K$ までのすべての整数をとり、 $d$ は1から $D$ までのすべての整数をとるとする。

#### 【0087】

置換行列 $Q(j, k)$ の大きさを $E$ として、 $Q(j, k)[e]$ で $Q(j, k)$ の $e$ 行目で1が立っている位置を表すとする。まず $Q(j, k)$ の行のインデックス $e$ を初期設定( $e=1$ )する (ステップ501)。

#### 【0088】

次に、アドレス発生手段102の $AG(k)$ では $Q(j, k)[e]$ を発生する (ステップ502)。

#### 【0089】

続いて、変数ノード関数手段103の $VN(k, d)$ はメモリ手段 $M(k, d)$ から $Q(j, k)[e]$ に基づいて決定されるアドレスの受信値、メッセージを読み込み、チェックノード関数手段105へのメッセージを生成する。また、直前のステップでチェックノード関数手段で生成されたメッセージを用いて $M(k, d)$ における直前のステップで指定されたアドレスの内容を更新する (ステップ503)。

#### 【0090】

次に $VN(k, 1)$ から $VN(k, D)$ で生成されたメッセージをシャッフルネットワーク手段 $SN(k)$ を介してチェックノード関数手段105へ送出する (ステップ504)。

#### 【0091】

$SN(k)$ は、置換行列 $P(j, k)$ で決定される順序で変数ノード関数手段とチェックノード関数手段とを連結している。

#### 【0092】

次に、チェックノード関数手段ではメッセージを生成する (ステップ505)。ここで $CN(d)$ は各エッジに対してメッセージを生成する。

#### 【0093】

$CN(d)$ で生成されたメッセージは、 $SN(k)$ に基づき変数ノード関数手段へ返される (ステップ506)。

#### 【0094】

次に、 $e$ を1インクリメントして $e=E+1$ となったら $AG(k)$ が $Q(j,k)$ に、 $SN(k)$ が $P(j,k)$ に基づく処理は終了であると判定する（ステップ507、508）。

【0095】

そうでないときにはステップ502に戻り処理を継続する。ステップ501から507までの処理はパイプライン化して行うことが可能である。

【0096】

次に、本発明の第2の実施の形態について図面を参照して詳細に説明する。

【0097】

本発明における大きな特徴の一つはシャッフルネットワーク手段105が $SN(1)$ から $SN(K)$ までの複数の局所的なシャッフルネットワークで構成されている点である。

【0098】

なお、かつアドレス発生手段102もシャッフルネットワーク手段105と同一の単位でメモリ手段へのアドレスを発生しているために、この構成単位に基づいた任意の並列度で実装することが可能となる。つまり図3のパリティ検査行列における列ブロックの個数に依らずに図1において $K$ の大きさを適切に設定して、並列度と装置複雑度を調節した実装を行うことが可能となる。

【0099】

本発明における第2の発明の実施の形態は図1で特に $K=1$ の場合を示しており、図6を参照するとメモリ手段601（ $M(1), \dots, M(D)$ ）と、アドレス発生手段602（ $AG$ ）と、変数ノード関数手段603（ $VN(1), \dots, VN(D)$ ）と、シャッフルネットワーク手段604（ $SN$ ）と、チェックノード関数手段605（ $CN(1), \dots, CN(d)$ ）で構成される。アドレス発生手段602は1個の回路で共有し、図1の102と比較してより多くのアドレスパターンを生成できるようにする必要がある。同様にシャッフルネットワーク手段604は1個の回路からなり、図1の104と比較してより多くの置換を表すことが可能であるようにする必要がある。チェックノード関数手段205は図1の105と比較して一度の入力数が小さくなるためにより簡単な構成とすることも可能である。

【0100】

図7は図3のパリティ検査行列を持つLDPC符号に対して図6の復号器を適用したときの処理の詳細な流れ図である(図5、図6との対比)。図7でjは図3のパリティ検査行列の行方向のブロックのインデックス、kは列方向のブロックのインデックスを表す。

【0101】

また、eは図3の $Q(j,k)$  (大きさE)の行のインデックスを表す。まずjを初期設定( $j=1$ )する(ステップ701)。次にeを初期設定( $e=1$ )する(ステップ702)。

【0102】

次に、kを初期設定( $k=1$ )する(ステップ703)。設定されたj,kに基づいてアドレス生成手段602であるAG、シャッフルネットワーク手段604であるSNをそれぞれ $Q(j,k)$ 、 $P(j,k)$ に設定する(ステップ704)。

【0103】

次にAGで $M(1), \dots, M(D)$ のアドレス $Q(j,k)[e]$ を発生する(ステップ705)。

【0104】

次に、変数ノード関数手段603の $VN(d)$ において読み込んだ情報に基づいてメッセージを生成するとともに、以前にチェックノード関数手段605で生成されたメッセージを用いて $M(d)$ の内容を更新する(ステップ706)。

【0105】

次に、ステップ706で生成されたメッセージを、SNを通してチェックノード関数手段605へ送出する(ステップ707)。

【0106】

次に、チェックノード関数手段605の $CN(d)$ においてメッセージ更新処理を行う(ステップ708)。ここで図5のステップ505と異なり、 $CN(d)$ では全入力揃っていないため、各ステップではメッセージ生成の途中の段階までを行っておくことになる。次にSNを通じて、 $CN(d)$ で以前に生成されたメッセージの中で生成されたものを変数ノード関数手段に送出する(ステップ709)。これがステップ706の $M(d)$ 更新のためのメッセージとなる。

【0107】

次に、列方向のインデックスを更新して、列方向の処理が終了しているかどうかを判定する（ステップ710，711）。

【0108】

終了している場合にはCN(d)においてメッセージを生成する（ステップ712）。

【0109】

次に、 $Q(j,k)$ のインデックスを更新して、 $Q(j,k)$ の次の列についてメッセージ更新を行うかどうかを判定する（ステップ713，714）。

【0110】

図7のステップ715、716、717は図4の404、405、406と同一である。

【0111】

【実施例】

次に、具体的な実施例を用いて本実施の形態の動作を説明する。

【0112】

復号の前に符号化の方法であるが、これは図8に示すように、情報Uに対してパリティ検査行列に乗じて0になるように冗長Pを付加する処理を符号器801で行う。

【0113】

このときに使用する行列は復号器で使用するHそのものではなく、符号化が容易になるように列を入れ替えた行列H' が用いられることがある。

【0114】

この場合には復号器の順序になるようにインターリーブ803を介して送信する。

【0115】

次に図3のパリティ検査行列の具体例を図9に示す。図9は $J=2$ 、 $K=3$ の場合で、パリティ検査行列構成要素901は $Q(j,k)$ の例を示している（大きさ $E=2$ ）。

【0116】

901において例えば $Q(2,1)$ は左下の $2 \times 2$ の部分行列、 $Q(1,3)$ は右上の $2 \times 2$ の

部分行列に対応する。パリティ検査行列構成要素 9 0 2 の  $P(j,k)$  の例についても同様である（大きさ  $D=3$ ）。パリティ検査行列 9 0 3 は 9 0 1 の  $Q(j,k)$  と 9 0 2 の  $P(j,k)$  のクロネッカー積で表される置換を部分行列とするパリティ検査行列である。

#### 【 0 1 1 7 】

以下、図 9 のパリティ検査行列に基づいて図 1 の復号器の実施例を述べる。

#### 【 0 1 1 8 】

図 1 0 (a) はメモリ手段 1 0 1 の構成例を示しており、 $M(k,d)$  の各エントリが何番目の符号語ビットのデータを保持しているかを示している。例えば  $M(1,1)$  は第 1、4 番目の符号語ビット、 $M(2,1)$  第 7, 10 番目の符号語ビットに対応するデータを保持していることを示している。一般に  $M(k,d)$  は  $D$  おきの  $E$  個の符号語ビットに対応するデータを保持している。図 1 0 (b) は  $M(k,d)$  の一つのエントリの構造を示している。

#### 【 0 1 1 9 】

図 9 のパリティ検査行列のグラフでは変数ノードのエッジの個数は 2 のため、2 個のメッセージ  $c(1), c(2)$  と受信値から構成されている。

#### 【 0 1 2 0 】

以下、図 5、6 の処理の流れを図 9 のパリティ検査行列の場合について述べる。

#### 【 0 1 2 1 】

最初にステップ 4 0 1 で  $j=1$  に設定され、 $AG(1), AG(2), AG(3), SN(1), SN(2), SN(3)$  は、それぞれ  $Q(1,1), Q(1,2), Q(1,3), P(1,1), P(1,2), P(1,3)$  に基づいて設定される。

#### 【 0 1 2 2 】

次にステップ 5 0 1 で  $e=1$  に設定され、 $AG(1), AG(2), AG(3)$  でアドレスが発生される。

#### 【 0 1 2 3 】

図 9 の 9 0 1 の  $Q(j,k)$  は 2 種類の置換から構成されており、 $(j,k)=(1,1), (1,3), (2,2)$  については式[式 4] というアドレスが発生する。



$$[式4] \quad Q(j, k)[1] = 1, \quad Q(j, k)[2] = 2$$

また、 $(j, k) = (1, 2), (2, 1), (2, 3)$ については、式[式5] というアドレスを発生する。

$$[式5] \quad Q(j, k)[1] = 2, \quad Q(j, k)[2] = 1$$

よって $j=1, e=1$ で $AG(1), AG(2), AG(3)$ で発生されるアドレスは[式4]、[式5]からそれぞれ1, 2, 1である。このアドレスに対する符号語ビットのインデックスは図10から順に1, 2, 3, 10, 11, 12, 13, 14, 15となる。当然であるが、図9の903の行列で最初の3行で要素が1である列番号に対応していることに注意する。

【0 1 2 4】

第 $m$ 番目の符号語ビットに対応する図11のメッセージ、受信値を $c(m, 1), c(m, 2), r(m)$ で表すとする。このとき $VN(k, d)$ で[式1]に基づいて生成されるメッセージ $v(k, d)$ は次のようになる。

[式6]

$$\begin{aligned} VN(1, 1) : v(1, 1) &\leftarrow r(1) + c(1, 2) \\ VN(1, 2) : v(1, 2) &\leftarrow r(2) + c(2, 2) \\ VN(1, 3) : v(1, 3) &\leftarrow r(3) + c(3, 2) \\ VN(2, 1) : v(2, 1) &\leftarrow r(10) + c(10, 2) \\ VN(2, 2) : v(2, 2) &\leftarrow r(11) + c(11, 2) \\ VN(2, 3) : v(2, 3) &\leftarrow r(12) + c(12, 2) \\ VN(3, 1) : v(3, 1) &\leftarrow r(13) + c(13, 2) \\ VN(3, 2) : v(3, 2) &\leftarrow r(14) + c(14, 2) \\ VN(3, 3) : v(3, 3) &\leftarrow r(15) + c(15, 2) \end{aligned}$$

シャッフルネットワーク手段 $SN(1), SN(2), SN(3)$ は、902の $P(1, 1), P(1, 2)$

,  $P(1, 3)$  の置換行列に基づくため、チェックノード関数手段  $CN(1), CN(2), CN(3)$  への入力は次のようになる。

$CN(1): v(1, 3), v(2, 1), v(3, 2)$

$CN(2): v(1, 2), v(2, 2), v(3, 3)$

$CN(3): v(1, 1), v(2, 3), v(3, 1)$

$CN(1), CN(2), CN(3)$  では [式 2] に基づいてメッセージが生成される。

【 0 1 2 5 】

ここで生成されるメッセージは [式 6] から次のものである。

[式 7]

$CN(1): c(3, 1), c(10, 1), c(14, 1)$

$CN(2): c(2, 1), c(11, 1), c(15, 1)$

$CN(3): c(1, 1), c(12, 1), c(13, 1)$

ここで生成されたメッセージは  $SN(1), SN(2), SN(3)$  を介して送出され、 $M(k, d)$  を更新する。

【 0 1 2 6 】

次のステップである  $j=1, e=2$  ではステップ 5 0 8 の分岐から  $AG(k), SN(k)$  は、そのまま、 $M(k, d)$  でのアドレスが  $Q(j, k)[2]$  に対応する符号語ビットに関してのメッセージの処理を実行する。このとき、[式 6]、[式 7] は以下のようになる。

【 0 1 2 7 】

$VN(1, 1): v(1, 1) \leftarrow r(1) + c(4, 2)$

$VN(1, 2): v(1, 2) \leftarrow r(2) + c(5, 2)$

$VN(1, 3): v(1, 3) \leftarrow r(3) + c(6, 2)$

$VN(2, 1): v(2, 1) \leftarrow r(10) + c(7, 2)$

$VN(2, 2): v(2, 2) \leftarrow r(11) + c(8, 2)$

$VN(2, 3): v(2, 3) \leftarrow r(12) + c(9, 2)$

$VN(3, 1): v(3, 1) \leftarrow r(13) + c(16, 2)$

$$VN(3,2): v(3,2) \leftarrow r(14) + c(17,2)$$

$$VN(3,3): v(3,3) \leftarrow r(15) + c(18,2)$$

$$CN(1): c(6,1), c(7,1), c(17,1)$$

$$CN(2): c(5,1), c(8,1), c(18,1)$$

$$CN(3): c(4,1), c(9,1), c(16,1)$$

次のステップである  $j=2, e=1$  の場合はステップ 4 0 5 の分岐からステップ 4 0 2 に戻り、 $AG(1), AG(2), AG(3), SN(1), SN(2), SN(3)$  はそれぞれ  $Q(2,1), Q(2,2), Q(2,3), P(2,1), P(2,2), P(2,3)$  に設定される。このとき [式 6]、[式 7] は以下のようになる。

#### 【 0 1 2 8 】

$$VN(1,1): v(1,1) \leftarrow r(4) + c(4,1)$$

$$VN(1,2): v(1,2) \leftarrow r(5) + c(5,1)$$

$$VN(1,3): v(1,3) \leftarrow r(6) + c(6,1)$$

$$VN(2,1): v(2,1) \leftarrow r(7) + c(7,1)$$

$$VN(2,2): v(2,2) \leftarrow r(8) + c(8,1)$$

$$VN(2,3): v(2,3) \leftarrow r(9) + c(9,1)$$

$$VN(3,1): v(3,1) \leftarrow r(16) + c(16,1)$$

$$VN(3,2): v(3,2) \leftarrow r(17) + c(17,1)$$

$$VN(3,3): v(3,3) \leftarrow r(18) + c(18,1)$$

$$CN(1): c(6,2), c(8,2), c(16,2)$$

$$CN(2): c(4,2), c(9,2), c(18,2)$$

$$CN(3): c(5,2), c(7,2), c(17,2)$$

同様に  $j=2, e=2$  の処理を実行して 1 iteration 処理が終了する。

#### 【 0 1 2 9 】

以下、図 1 の構成要素のヴァリエーションについて述べる。

変数ノードの次数が高いときには図 11 に示すように、メモリ手段の各エントリにおいて  $r$  の代わりに  $r$  とメッセージの和を保持しておく方法も考えられる。図 11 では [式 1] の処理が一度の減算で済むため、変数ノードの次数が高い場合には計算量の削減を図ることができる。

#### 【0130】

次に、アドレス発生手段 102 については簡単な回路によってアドレスを発生することで装置複雑度を低減することができる。最も簡単な置換は巡回置換であり、これはカウンタを巡回的に用いて実現できる。カウンタの初期値を変更することで異なる置換を表すことができ、カウンタの周期を変えることで異なる符号長の符号の復号も実行可能となる。

#### 【0131】

アドレス発生手段を巡回置換で構成した場合にはシャッフルネットワーク手段 105 はそれよりは複雑な置換とすることで符号性能を向上させることができる。この中でも回路としての実装が容易なものとして、ガロア体の乗除算や指数・対数演算を用いる方法が挙げられる。

#### 【0132】

チェックノード関数手段 105 は MP 復号を min-sum アルゴリズムとすることで比較的実装が容易となるが、このとき復号性能が最良の場合である sum-product アルゴリズムに比べて劣化する。この問題に対してチェックノードで生成されるメッセージに対しては 1 より小さい正数を乗ずることで特性を改善することができる。

#### 【0133】

図 6 の第 2 の実施の形態において、チェックノード関数手段 605 は各ステップで 1 入力ずつ得ることになる。[式 2] の min-sum アルゴリズムを用いている場合には図 12 に示すようにチェックノード関数手段を簡易に実現することが可能となる。変数ノード関数手段からのメッセージは分離手段 1200 で絶対値部分と正負の符号部分に分離される。絶対値部分は二つの比較器 1201、1204 を用いて最小のもの二つを求める。レジスタ 1202 は最小の絶対値を、レジスタ 1205 は 2 番目に小さい絶対値を保持する。レジスタ 1203 は最小の絶対

値となるインデックスを保持する。一方、正負の符号部分はレジスタ1206に保持するとともに全体の排他的論理和もレジスタ1207に保持しておく。これだけの情報を保持すれば[式2]の出力を各エッジに対して生成することは容易に行うことができる。

#### 【0134】

以上、図3のパリティ検査行列ように行、列の1の個数が一定の正則なLDPC符号について実施例を示してきたが、本発明は非正則なLDPC符号の復号にも適用可能である。図4の $(P(j,k))$ ,  $(Q(j,k))$ においていくつかのブロックを零行列に置き換えることによって非正則なLDPC符号を構成することができる。

#### 【0135】

このとき図1、6に示した復号器を適用することができる。図1の復号器の場合、零行列に当たる部分に対しては0である確率が非常に高いと常に設定することで対処することができる。また、図6の復号器では零行列にあたる部分は単にスキップすればよい。

#### 【0136】

図13は優れた符号特性が得られる非正則なパリティ検査行列の例で、 $J=5, K=10$ の図4の行列において置換行列をいくつか零行列に置き換えたものである（ $P$ は置換行列、 $0$ は零行列）。本発明ではこのように置換行列を並べたパリティ検査行列を持つ符号の復号器として動作するため、与えられた $J, K$ の下で最適な非正則なパリティ検査行列の構成を求めることが容易となる。

#### 【0137】

尚、上述の第1の実施の形態及び第2の実施の形態における、アドレス発生手段102と、変数ノード関数手段103と、チェックノード関数手段104と、シャッフルネットワーク手段105との全部又は一部をプログラムで動作するCPUやMPUによって置き換えても良い。

#### 【0138】

##### 【発明の効果】

第1の効果は、アドレス発生手段102およびシャッフルネットワーク手段を複数の置換として実現し、かつアドレス発生手段を複数のメモリ手段に対応させ

ることで符号性能を保ったまま装置規模削減を図ることができることである。

#### 【 0 1 3 9 】

符号性能を保つことができる理由は、アドレス発生手段 1 0 2 とシャッフルネットワーク手段 1 0 5 における置換を同一周期で変更するために図 3 に示すパリティ検査行列を持つ LDPC 符号の復号に対応でき、このパリティ検査行列の中では性能の良い符号が構成できるためである。

#### 【 0 1 4 0 】

第 2 の効果は、並列度と装置規模のトレードオフの調節が可能な柔軟な装置構成を与えることができることにある。その理由は、シャッフルネットワーク手段 1 0 5 の各シャッフルネットワークが変数ノード関数手段 1 0 3 の一部のものと局所的に連結されているために、この単位の倍数で任意の並列度で実装しても回路を実行するスケジュールが容易に組めることにある。

#### 【図面の簡単な説明】

##### 【図 1】

本発明の第 1 の実施の形態の構成を示すブロック図である。

##### 【図 2】

第 1 の実施の形態の動作の流れ図である。

##### 【図 3】

第 1 の実施の形態に適用される LDPC 符号のパリティ検査行列の構成である。

##### 【図 4】

第 1 の実施の形態の図 3 のパリティ検査行列に対する動作を示す流れ図である。

##### 【図 5】

第 1 の実施の形態の図 3 のパリティ検査行列に対する動作を示す流れ図である。

##### 【図 6】

本発明の第 2 の実施の形態の構成を示すブロック図である。

##### 【図 7】

第 2 の実施の形態の動作の流れ図である。

## 【図 8】

本発明の復号装置に対する符号化装置のブロック図である。

## 【図 9】

第 1 の実施の形態に適用される LDPC 符号のパリティ検査行列の例を示す図である。

## 【図 1 0】

第 1 の実施の形態におけるメモリ手段の構成例を示す図である。

## 【図 1 1】

第 1 の実施の形態におけるメモリ手段の構成例を示す図である。

## 【図 1 2】

第 2 の実施の形態におけるチェックノード関数手段の構成例を示す図である。

## 【図 1 3】

本発明が適用される非正則な LDPC 符号のパリティ検査行列の例を示す図である。

## 【図 1 4】

LDPC 符号のパリティ検査行列とそのタナーグラフの例を示す図である。

## 【図 1 5】

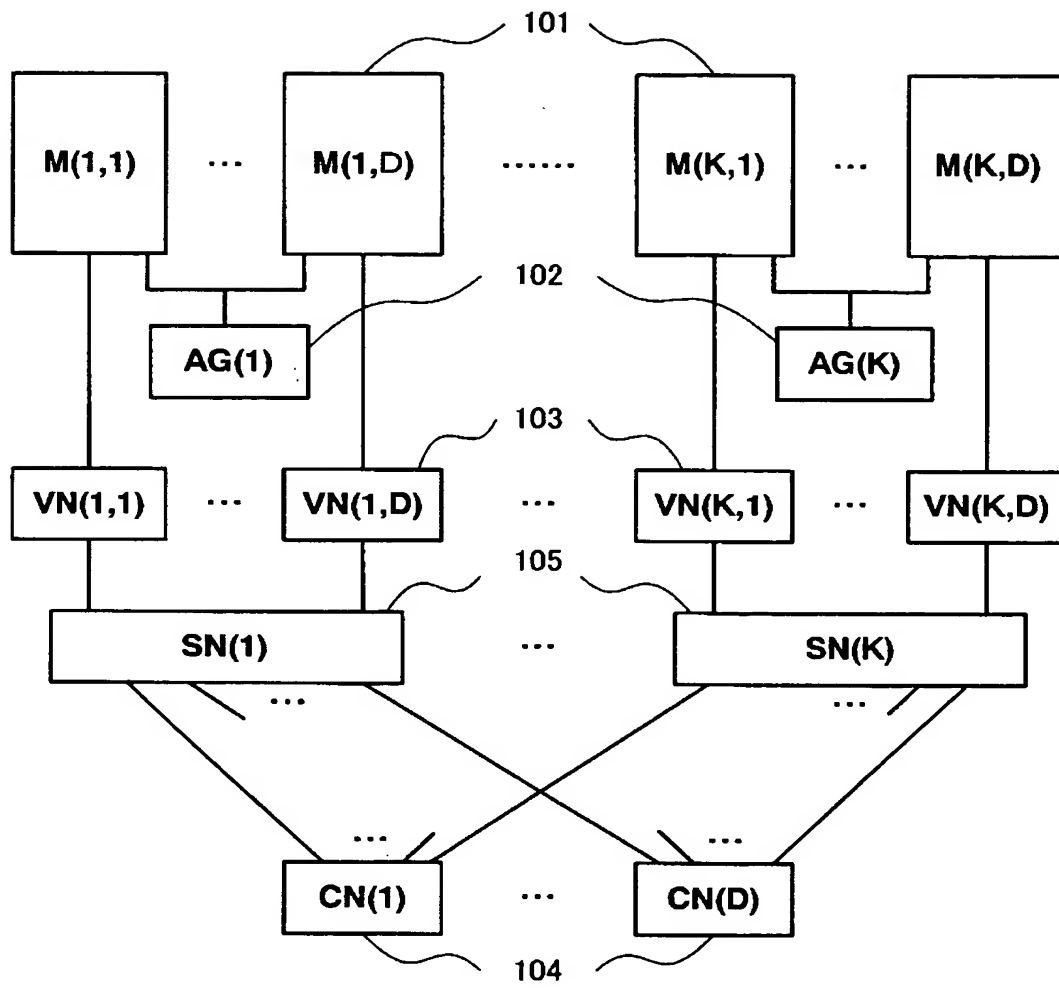
復号装置の従来例を示す図である。

## 【符号の説明】

1 0 1、6 0 1、1 5 0 1	メモリ手段
1 0 2、6 0 2、1 5 0 2	アドレス発生手段
1 0 3、6 0 3、1 5 0 3	変数ノード関数手段
1 0 4、6 0 5、1 5 0 4	チェックノード関数手段
1 0 5、6 0 4、1 5 0 5	シャッフルネットワーク手段

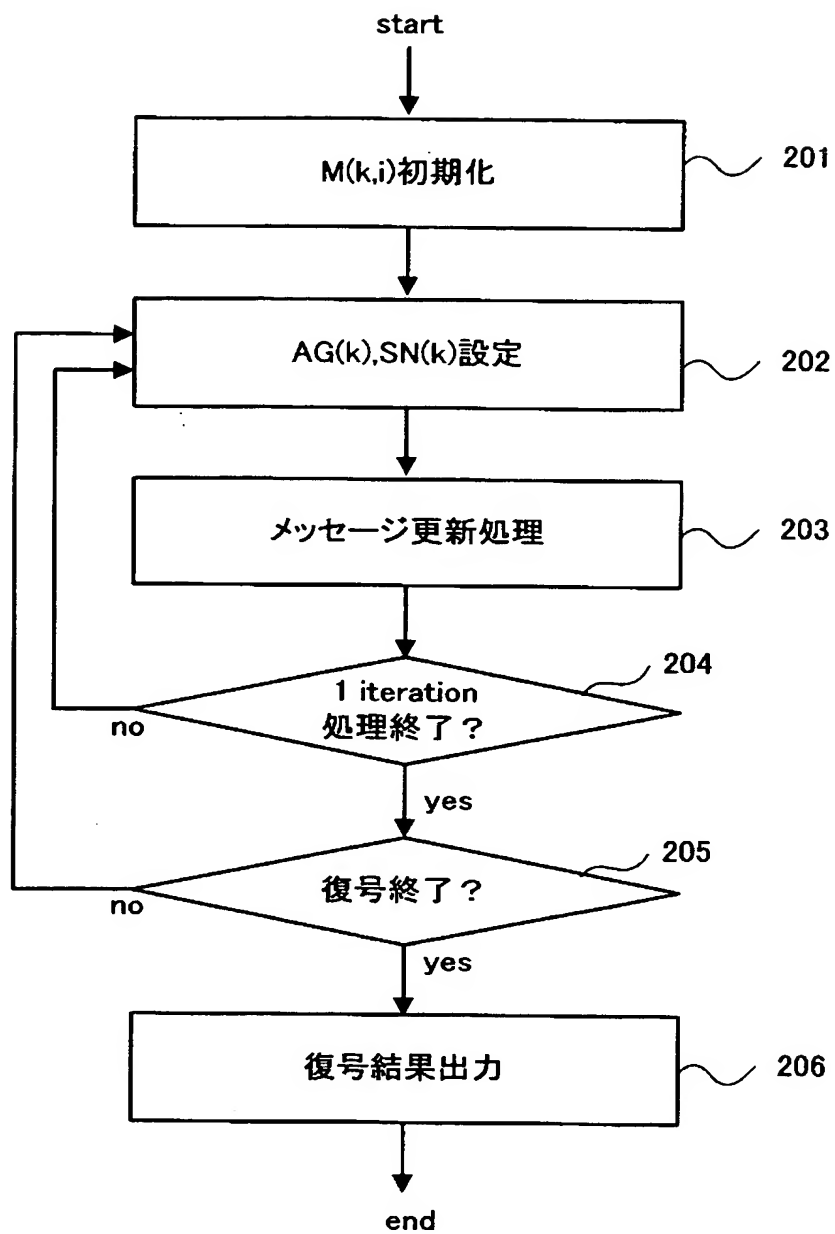
【書類名】 図面

【図 1】





【図 2】

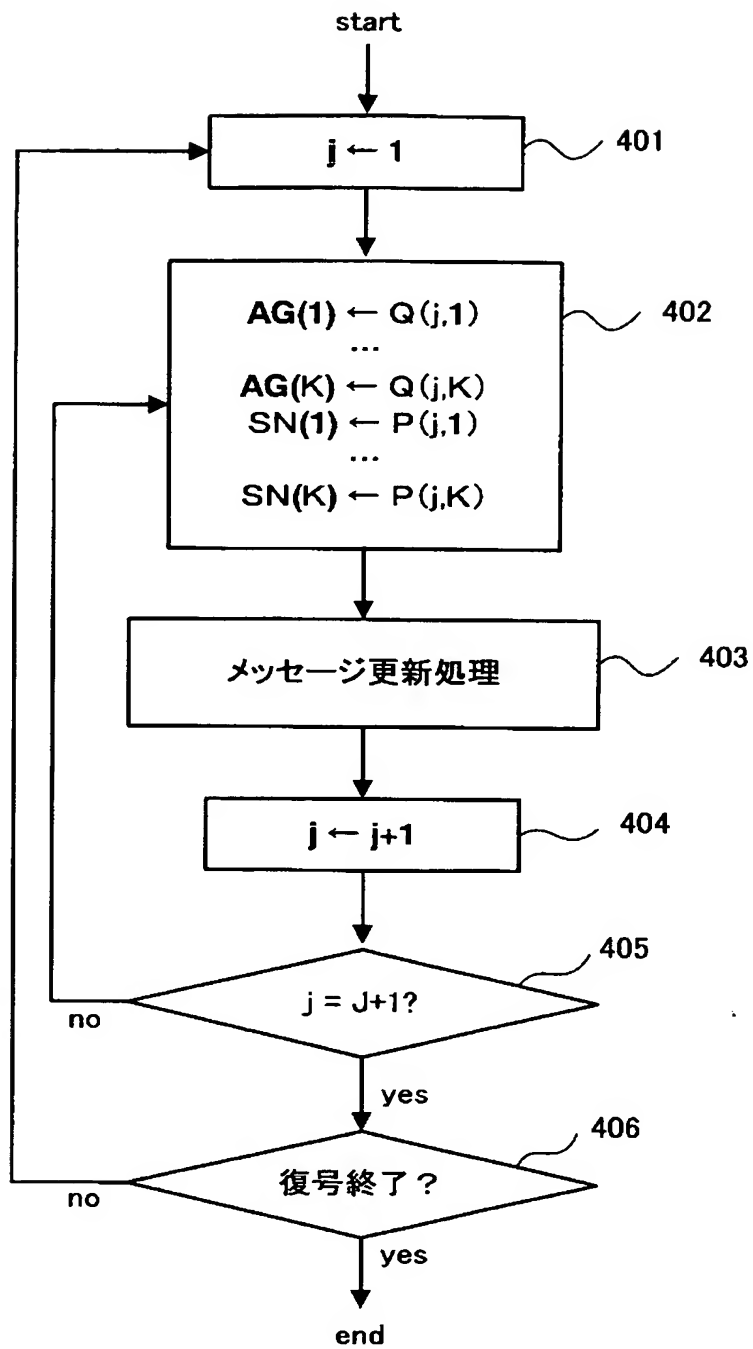


【図 3】

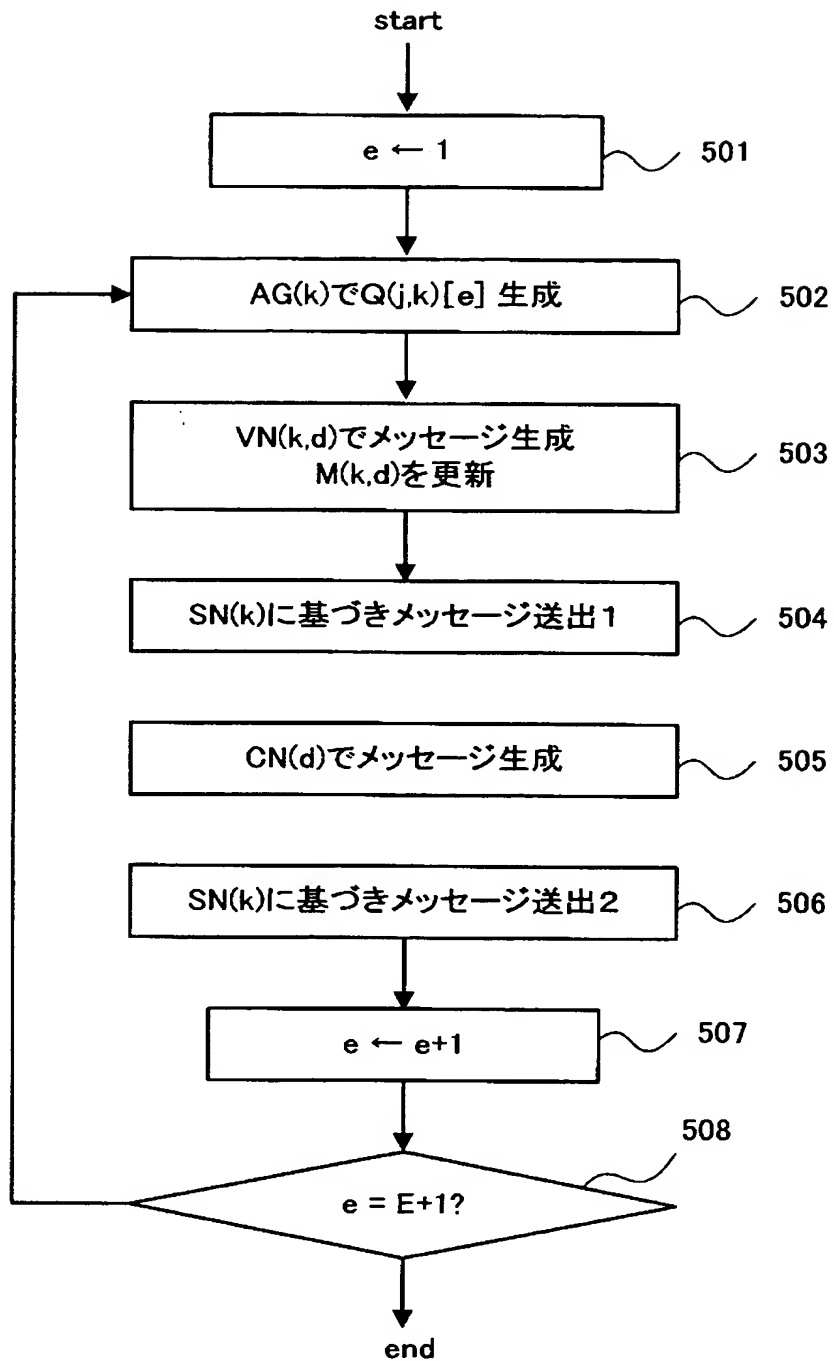
$$H = \begin{pmatrix} R(1,1) & \cdots & R(1,K) \\ \cdots & \cdots & \cdots \\ R(J,1) & \cdots & R(J,K) \end{pmatrix} \quad 301$$

$$R(j,k) = Q(j,k) \otimes P(j,k) \quad 302$$

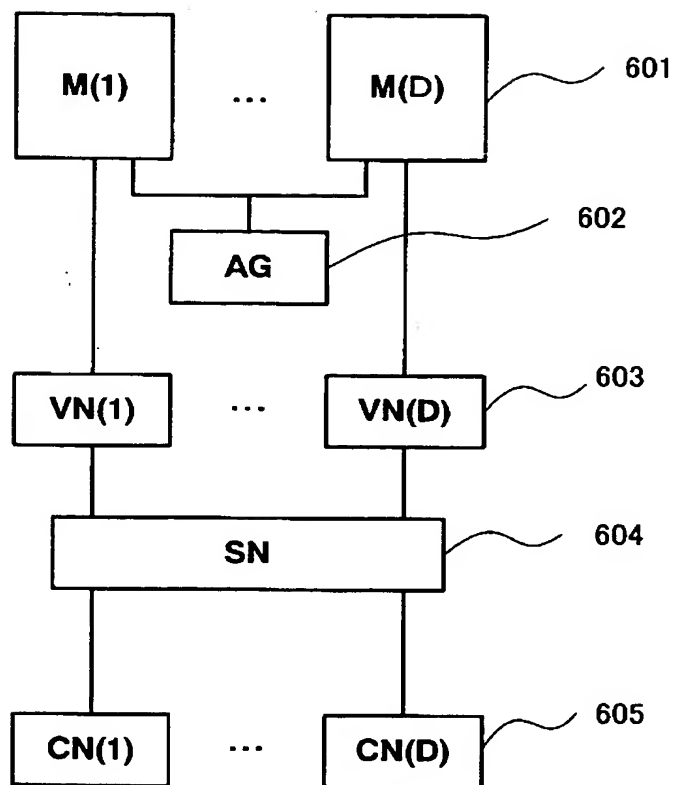
【図 4】



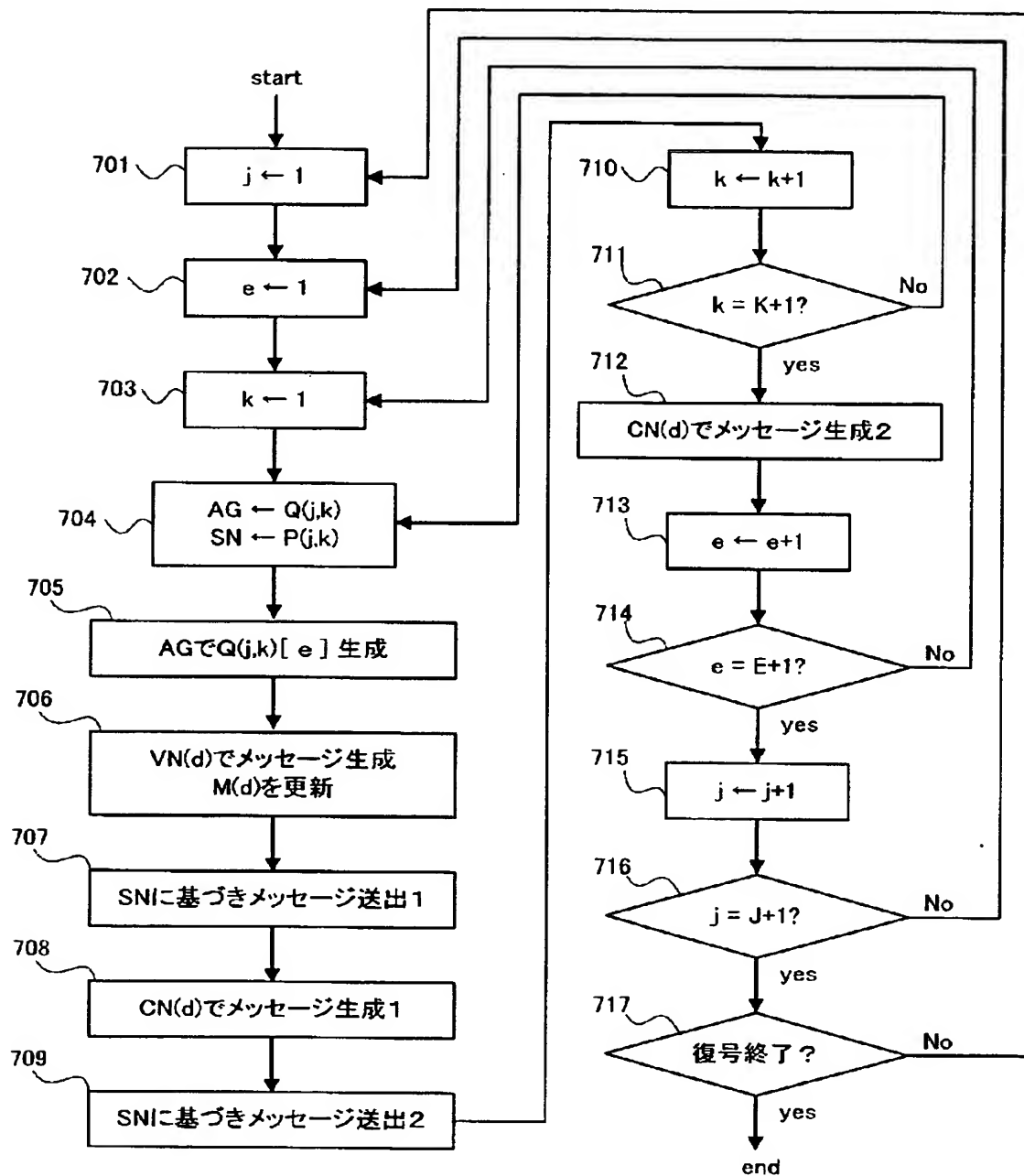
【図 5】



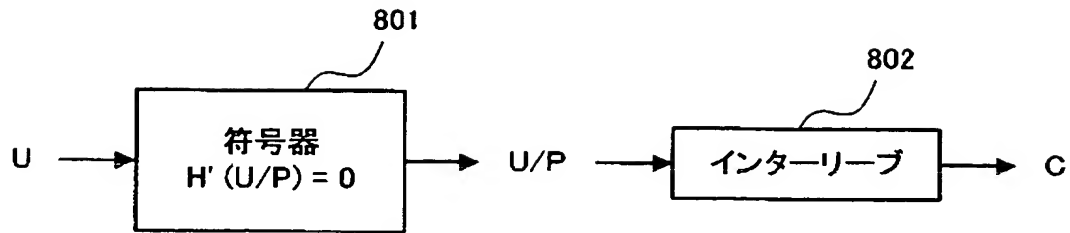
【図 6】



【図 7】



【図 8】



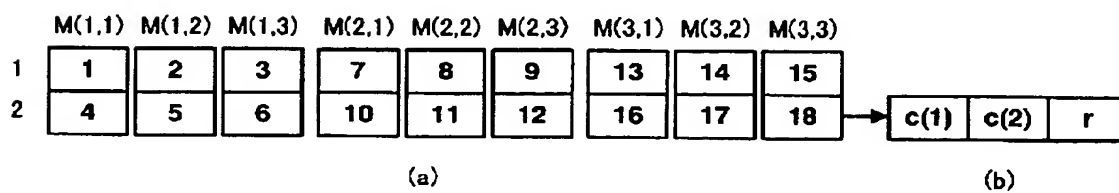
【図 9】

$$(Q(j,k)) = \begin{pmatrix} 10 & 01 & 10 \\ 01 & 10 & 01 \\ 01 & 10 & 01 \\ 10 & 01 & 10 \end{pmatrix} \quad (901)$$

$$(P(j,k)) = \begin{pmatrix} 001 & 100 & 010 \\ 01 & 001 & 0001 \\ 100 & 001 & 100 \\ 001 & 010 & 100 \\ 100 & 001 & 001 \\ 01 & 01 & 001 & 010 \end{pmatrix} \quad (902)$$

$$H = (R(j,k)) = (Q(j,k) \otimes P(j,k)) = \begin{pmatrix} 001000 & 000100 & 010000 \\ 010000 & 000010 & 001000 \\ 100000 & 000001 & 100000 \\ 000001 & 100000 & 000010 \\ 000010 & 010000 & 000001 \\ 000100 & 001000 & 000100 \\ 000001 & 010000 & 000100 \\ 000100 & 001000 & 000001 \\ 000010 & 100000 & 000010 \\ 001000 & 000010 & 100000 \\ 100000 & 000001 & 001000 \\ 010000 & 000100 & 010000 \end{pmatrix} \quad (903)$$

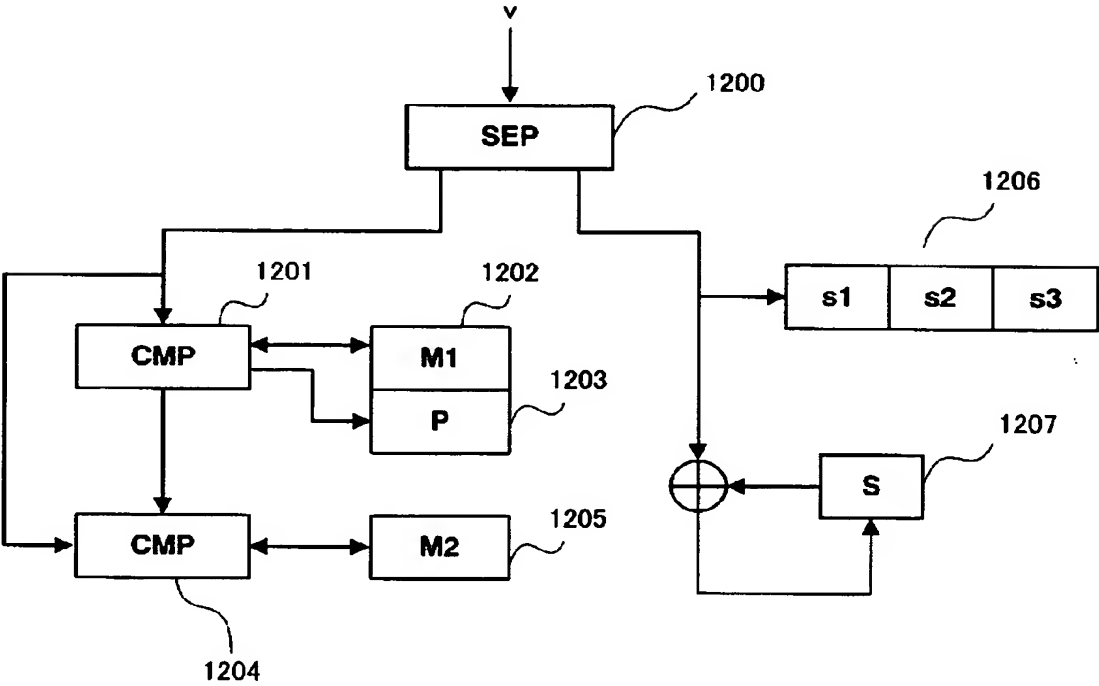
【図 10】



【図 1 1】

c(1)	c(2)	c(3)	c(4)	$r+c(1)+c(2)+c(3)+c(4)$
------	------	------	------	-------------------------

【図 1 2】



【図 1 3】

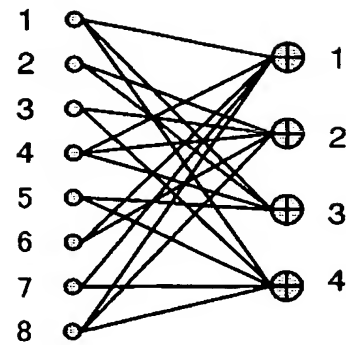
$$\begin{pmatrix} P & P & 0 & 0 & 0 & 0 & P & P & P & P \\ P & P & P & P & P & 0 & 0 & 0 & 0 & P \\ P & P & P & P & 0 & P & 0 & 0 & P & 0 \\ P & P & P & 0 & P & P & 0 & P & 0 & 0 \\ P & P & 0 & P & P & P & P & 0 & 0 & 0 \end{pmatrix}$$



【図 14】

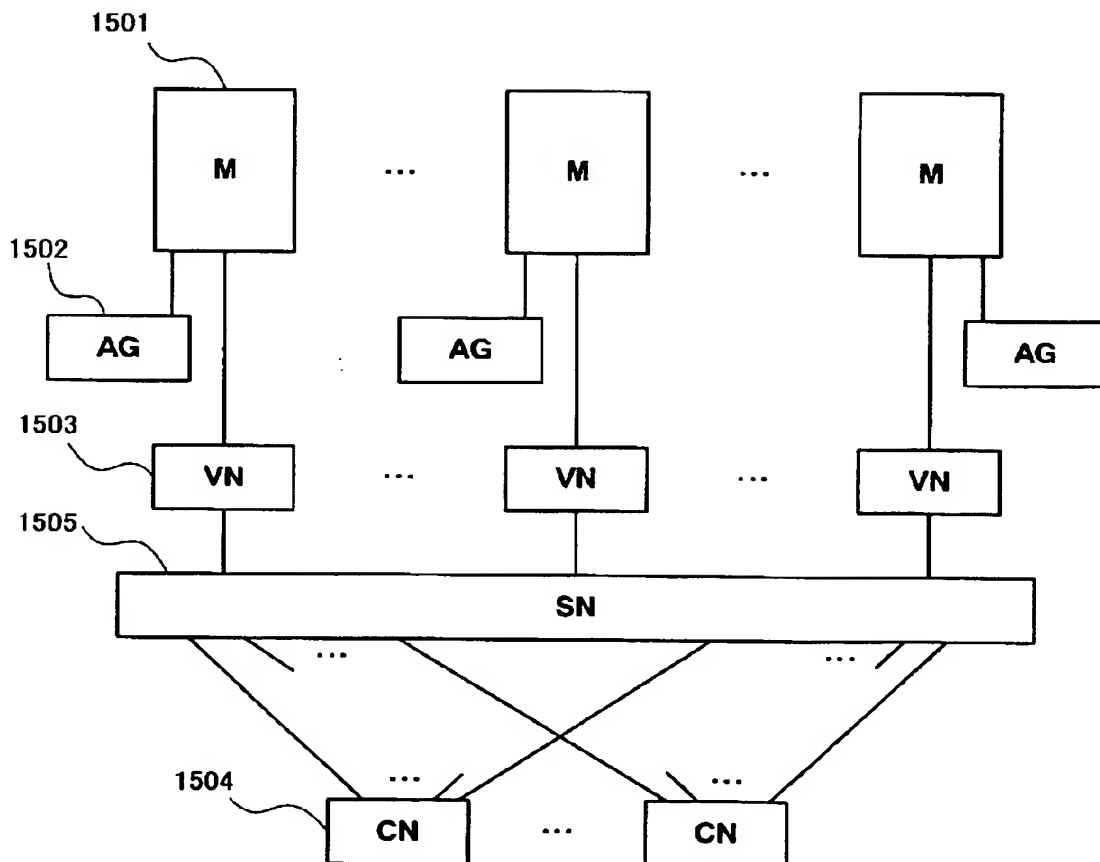
$$H = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}$$

(a) パリティ検査行列 H



(b) タナーグラフ G

【図 15】



【書類名】 要約書

【要約】

【課題】 LDPC符号の復号装置において従来装置と比べて簡易で、かつ柔軟度の高い並列復号器を提供する。

【解決手段】 受信値および記復号方式の過程で発生するメッセージを格納する複数のメモリ手段と、複数の変数ノード関数手段と、複数のチェックノード関数手段と、メモリ手段のアドレスを発生する複数のアドレス発生手段と、変数ノード関数手段とチェックノード関数手段の接続を定める複数のシャッフルネットワーク手段を備える。アドレス発生手段はアドレスを複数の置換に基づいて発生する。シャッフルネットワーク手段はそれぞれ一部の前記変数ノード関数手段と接続され、この接続は複数の置換に基づいて決定されている。復号過程におけるアドレス発生手段の置換の変更とシャッフルネットワーク手段の置換の変更は同一周期で行う。

【選択図】 図1

認 定 ・ 付 加 情 報

特許出願の番号	特願 2 0 0 2 - 3 5 0 8 4 6
受付番号	5 0 2 0 1 8 2 7 6 6 9
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 4 年 1 2 月 4 日

< 認定情報・付加情報 >

【提出日】	平成14年12月 3日
-------	-------------

次頁無

特願 2 0 0 2 - 3 5 0 8 4 6

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 4 2 3 7 ]

1. 変更年月日

1 9 9 0 年 8 月 2 9 日

[変更理由]

新規登録

住 所

東京都港区芝五丁目 7 番 1 号

氏 名

日本電気株式会社